Unsteady Aerodynamics of Aircraft Maneuvering at Angle of Attack

Kenneth E. Wurtzler, Robert F. Tomaro, Matthew J. Grismer, William Z. Strang
Air Force Research Laboratory/Air Vehicles Directorate
WPAFB, OH

Abstract

The focus of this project is the utilization of an unstructured, parallel CFD flow solver to predict the unsteady aerodynamics acting on a fighter configuration maneuvering at angles of attack. Emphasis is centered on the routine application of analysis methods (grid generation, flow solver, data visualization) to obtain accurate and timely results. Techniques used in the simulation of an F-18C and F-16C at high angles of attack are applied to an F-18 E/F at moderate angles of attack. Specifically to investigate an abrupt wing stall feature over one wing. Viscous, turbulent solutions at various static angles of attack have been completed and are presented. The flight conditions are Mach = 0.8 and 0.9 and angles of attack 8^o through 12^o. The results reveal the angle of attack where a region of separated flow develops on the wing and the subsequent loss of lift for the aircraft. A comparison of two turbulence models (Spalart-Allmaras and Menter' s SST) and their effect on the calculation of the stall angle of attack is discussed.

Introduction

The state of the art in numerical simulation of aircraft is generally limited to steady-state analysis of complete aircraft or unsteady analysis of simple geometries or aircraft components. Grid generation and geometry modeling has matured to the point where aircraft with weapons and detailed geometric complexity can be modeled. Further advancement to an analysis of viscous, unsteady aerodynamics of a complete aircraft requires large amounts of computer time due to the large number of grid points and the slow advancement of the solution process.

Research into the area of high angle of attack aerodynamics has been dominated by experimental testing. Wind tunnel tests have been used initially to investigate the impact of different forebody shapes

nose, even at zero sideslip, can dominate directional stability, creating large yawing moments. These small side forces are a result of asymmetrical shedding of the forebody vortices. Small surface imperfections such as radome gaps, dents, and sharp paint depth mismatches can affect the strength and path of one of the vortices. The resultant net side force can then increase and the flow becomes unstable. This condition of aircraft experiencing severe yawing moments at high angle of attack flight is called nose-slice departure. It is an unsteady phenomenon, which can be catastrophic.

Recent progress in the use of computational fluid dynamics has shown that an accurate solution can be obtained in these cases [1]. A wind tunnel analysis of surface imperfections and the addition of various strakes to the forebody was conducted to assess possible flow control methods. The computational analysis utilized $Cobalt_{60}$ to compute the forebody vortices and impact of surface imperfections and vortex control methods. Comparisons to wind tunnel results showed very good agreement and concluded the applicability of numerical methods to successfully model the asymmetrical vortex shedding and flow control methods.

The advent of high performance computing has opened this application area to full aircraft. The majority focus of the 2-year DoD Challenge Project entitled "Unsteady Aerodynamics of Aircraft Maneuvering at High Angle of Attack" has been the application of $Cobalt_{60}$ to high angle of attack flows. The results from these calculations have aided the understanding of proper geometry handling, grid generation, turbulence model requirements, and flow physics. These same features are now being used for an analysis of the F-18 E/F at moderate angles of attack. This aircraft has exhibited the features of an abrupt wind stall on one wing during a maneuver. The operational aspect of this aircraft has been corrected but further research was warranted to try to gain an understanding A combined wind tunnel and computational approach has been taken to further the understanding of the causes and development of this feature.

Technical Approach

The technical approach involves the seamless application of grid generation, flow solver, and data visualization. Grid generation was accomplished with the code VGRIDNS from NASA Langley. Flow solutions were calculated using $Cobalt_{60}$. Flow visualization was achieved with the code FieldView. The first phase of the project consisted of steady-state calculations at angles of attack 7, 9, 11, and 12 at a Mach number of 0.8. The first grid modeled the wing with a thick trailing edge while the second grid modeled the trailing edge as infinitely thin. In actuality, any edge on an aircraft has some thickness to it. The computer model can create a razor-sharp edge that might be sharper than exists. The horizontal and vertical tails were not modeled to match the experimental set-up.

was done to make the grid generation of the boundary layer easier. Even though the wind tunnel model or flight aircraft might be defined as having a sharp trailing edge, it will actually have some thickness.

The second grid consisted of 5.31 million cells. Of those cells, 4.44 million cells were tetrahedrons and 874,816 cells were prisms. The surface was modeled with 109,352 triangles with the far field boundaries containing 300 triangles and the symmetry boundary containing 22,282 triangles. The average y+ for the grid was 0.25. The trailing edge of the wing was modeled as being infinitely thin.

The process to obtain the solutions began with a steady run at 7^o angle of attack. This was run to convergence, which occurred between 4,000 and 6,000 iterations depending on the Mach number. For the 9^o case, the grid was then statically rotated 2^o and the solution was restarted from the converged 7^o case. The convergence took between 2,000 and 3,000 iterations. Each successive angle of attack was calculated in the same manner. The convergence history for the Mach = 0.8 case for three angles of attack is shown in Figure 1. The convergence for the first angle of attack occurred at 4,500 iterations. The longer convergence of the body forces for 11^o reveals the slow development of the separated region over the wing. The steady convergence of y+ reveals that the boundary layer development is not interrupted by the grid rotation. Flight conditions used for these calculations were Mach = 0.8 and 0.9. The cases were run with the Spalart-Allmaras turbulence model and Menter's SST model.

Flow Solver Methodology

$Cobalt_{60}$ is a parallel, implicit unstructured Euler/Navier-Stokes flow solver. $Cobalt_{60}$ accepts arbitrary cell types and allows a variety of cell types in a single grid. The finite-volume, cell-centered solver is based on the exact Riemann solver of Godunov. A more efficient Riemann solver based on Colella's work coupled with the iterative method of Gottlieb and Groth is utilized.

Second-order spatial accuracy is achieved through a least squares reconstruction of the primitive variables. The least squares influence matrix is composed of the unknown coefficients of a second-order accurate Taylor series using the nearest neighbors of the current cells as a stencil. This method approximates a "central-differenced" gradient at the cell face. In general, the influence matrix is over-specified since the current cell has more neighboring cells than unknown coefficients. Thus, the "fit" of the data with the minimum L_2 norm error is solved for by a QR factorization. The QR factorization is chosen over an LU elimination because of its greater stability, which is important on high aspect ratio viscous meshes. In general, the face values must be limited so as to be non-extremal. Limiting is an important task, as stability of the method is required without being overly dissipative, thereby negating

component is constructed as a function of the cell centroid values and the tangential component. Turbulent solutions are obtained by using the Spalart-Allmaras turbulence model.

Implicit integration schemes are unconditionally stable, and therefore allow an arbitrary time step to be taken. Realistically, there is a physical time step limit for each unsteady case. If this time step limit is violated, the algorithm will provide answers but will miss some of the important unsteady flow characteristics. This limit depends on the time scales of the unsteady flow phenomena being simulated. The implicit temporal integration scheme was reported by Tomaro *et al* [3]. Second-order temporal accuracy is achieved by a one point backwards Taylor series expansion in time. $Cobalt_{60}$ employs a cell-by-cell implicit scheme including both the analytical inviscid and viscous Jacobians, thereby linearizing the governing equations. The inviscid Jacobians are split according to Jameson and Turkel to accurately capture the flow propagation and to increase stability. The inviscid splitting has been modified to eliminate upstream propagation of information in supersonic and hypersonic flows. The viscous Jacobians are implemented for robustness of turbulent solutions and are split on a simplified eigenvalue approach. The implicit algorithm couples the flow equations and allows for faster convergence rates. Typically, steady-state solutions are converged under 2,000 iterations. The coupled equations create an influence matrix for each cell that must be inverted. This is accomplished by be employing a LU decomposition on the matrix equation resulting in a Gauss-Seidel iterative scheme. Multiple matrix sweeps can be employed to obtain a better solution to the linearized equations. A Newton sub-iteration method is utilized to better converge each time step in an unsteady problem, thereby allowing larger time steps.

Turbulence Models

The inclusion of the Spalart-Allmaras and Menter' s SST (Shear-Stress Transport) turbulence models into $Cobalt_{60}$ are left to a previously published paper [4]. The focus of their participation in this effort is to see their limitations for separated flows.

The Spalart-Allmaras model is a one-equation turbulence model that is local (i.e. the equation at one point does not depend on the solution at other points). It can therefore be used on a grid of arbitrary structure. It is designed to work well for flow over wings.

Menter' s SST model is based on a combination of the two-equation κ-ω model of Wilcox and a standard κ-ε model [3]. In the inner 50% of the boundary layer, this model is identical to the Wilcox model but changes gradually to the standard κ-ε model towards the boundary-layer edge. This model is also virtually identical to the κ-ε model for free shear layers. Menter' s modification accounts for the

arrays are allocated at runtime, eliminating the need to recompile the code for different grids/problems. This also makes it possible to reduce memory requirements when the user requests simpler physics or numerics. Array syntax and array operations replace do loops where appropriate, making the code more readable. Global variables are defined in modules, replacing, replacing many named common blocks, and a single easy-to-change parameter was defined to globally set the code to single- or double-precision numerical accuracy. Overall, using Fortran 90 significantly improves software modularity and maintainability and makes it possible to use one source code for all parallel platforms.

Parallelization follows the domain decomposition paradigm where each processor operates on a subsection (zone) of the original grid. The Message Passing Interface (MPI) library is used to pass information between processors. Various MPI features, such as non-blocking communications, persistent communication requests, and vector/indexed MPI datatypes are used to minimize the communications overhead. Considerable care was taken to ensure the zonal boundaries are computed in exactly the same manner as the zonal interiors. At zonal boundaries, the conserved variables for each boundary cell, the initial conditions for the Riemann problem for each boundary face and the gradients in each boundary cell (viscous cases only) are passed to the neighboring zones (processor). Additionally, the implicit time integration scheme requires the communication of boundary cell residual data during each sweep of the iterative matrix solver. Since the Jacobi method uses residuals from the previous sweep only, these residuals are communicated at the end of each sweep. The symmetric Gauss-Seidel method, however, uses the latest residuals as they become available. Technically, then, the residuals of zonal boundary cells should be communicated to neighboring zones (processors) immediately upon their calculation. However, we have found that passing the zonal cell residuals at the end of each sweep works very well with the Gauss-Seidel method. This introduces some Jacobi character at the zonal boundaries, which theoretically slows the convergence rate for multi-zonal (parallel) cases. However, we have experienced virtually no degradation in convergence rate, certainly because the interface sizes have been kept relatively small. Even if a noticeable degradation in convergence were to exist, additional sweeps and/or additional Newton sub-iterations would remove the problem. A discussion of these two zonal boundary treatments is given in Grismer [5].

The major unstructured grid-generation systems that we use, VGRIDns and TETMESH [xx], create single-zone meshes exclusively. Therefore, the Parallel METIS domain decomposition library of Karypis and Kumar [6,7], has been directly incorporated into the flow solver to divide the grid into nearly equally sized zones that are then distributed one per processor. METIS is used because of its demonstrated performance in the rapid generation of nearly equally balanced zones (based on cell number) with minimal interface size. Care is taken to evenly distribute the storage requirements of the original, single-zone grid across all the processors during this process. This maintains the memory

overhead, explaining its scalability as described below. Details about the multi-level recursive bisection techniques used by METIS, as well as its performance versus other methods, are contained in [6].

Figure 3 indicates the scalability of the code on four different architectures, an IBM SP2, IBM SP3, Cray T3E, and an SGI Origin2000. The figure shows the speedup obtained over one processor versus the number of processors for a given problem. The benchmark problem considered is a three-dimensional, inviscid flow ($M_\infty = 0.85$) over an arrow-wing body configuration, modeled with 435,000 tetrahedra. The ideal line represents ‘perfect’ scalability. Modified Amdahl’s line takes into account cache efficiency as the problem on each processor becomes smaller. As one approaches a large enough number of processors, other issues such as communication override the benefits of improved cache efficiency. The “superscalability” is the result of improved cache performance on each processor as the size of the grid zones decreases. The randomness of unstructured grids results in poor cache utilization: a byproduct of partitioning the grid is better data locality and therefore improved cache performance.

HPCMP Resources

The computations achieved during this project were acquired using the Maui HPCC (MHPCC) IBM SP2 and the NAVO T3E. For FY00, this project had 300,000 hours assigned for the MHPCC IBM SP2 and 300,000 hours for the NAVO T3E.

Findings/Insights

The initial comparison between a ‘thick’ trailing edge and an infinitely thin trailing edge is discussed. For all the computations, only one half of the aircraft was modeled with no horizontal and vertical tails. Several views of the domain discretization are shown in Figure 4. The leading edge of the wing on the F-18 E/F contains a snag mid-chord as is evident in the figure. The flaps were set according to the wind tunnel and flight test conditions. The leading edge flap was down 10^o, the inboard trailing flap down 10^o, and the outboard flap down 5^o.

The initial part of the computational effort was devoted to the proper modeling of the trailing edge of the wing. This geometric choice impacts the unstructured grid generation. The development of the flow over the wing is show in Figure 5 for angles of attack 7^o, 9^o, 10^o, and 11^o for the thick trailing edge case. The angle of attack at which the separation occurs is predicted to be between 10^o and 11^o. The region of separated flow develops over the flaps and moves forward over the flat mid-section of the wing after 10^o is attained. The infinitely thin trailing edge case is shown in Figure 6 for angles of attack

differences. The thick trailing edge makes for easier grid generation, but in this case it impacts the separation angle of attack. The experimental effort also investigated the two different trailing edges and their separation data showed the opposite effect. The thick trailing edge delayed separation. Their model had the horizontal and vertical tails on during those runs though. The infinitely thin trailing edge was chosen for the experimental model and was used for the remainder of the computational effort. The data from this portion of the effort was used as a baseline for future comparisons.

At Mach = 0.8, similar separation data is shown in Figure 8. At this lower Mach number, the separation occurs at 8^o. Lift curve data is shown in Figure 9. The separation bubble is able to move upwing at a lower angle of attack due to lower greater adverse pressure gradient.

The result of using Menter' s SST turbulence model is shown in Figure10 and compared to the Spalart-Allmaras turbulence data in Figure 11. This comparison is for Mach = 0.9. The SST model captures the adverse gradient better and therefore causes the separation to occur earlier than the Spalart-Allmaras model data. This more closely matches the experimental results. The critical physical feature is the development of the adverse pressure gradient over the wing. An accurate capture of that will yield a very good flow field description.

Acknowledgements

The work mentioned in this paper was supported by the efforts of the following people: Hugh Thornburg and Frank Witzeman. The support of the NAVO MSRC and the Maui HPCC is greatly appreciated.

References

1. Wurtzler, K., "An Effectiveness Study of F-15 Forebody Flow Analysis Using $Cobalt_{60}$," AIAA 99-0536, January 1999.
2. Strang, W.Z., Tomaro, R.F., and Grismer, M.J., "The Defining Methods of *$Cobalt_{60}$*: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver," AIAA 99-0786, January 1999.
3. Tomaro, R.F. and Strang, W.Z., "An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Grids," AIAA 97-0333, January 1997.
4. Menter, F.R., "Zonal Two Equation κ-ω Turbulence Models for Aerodynamic Flows," AIAA-93-2906, July 1993.
5. Grismer, M.J., Strang, W.Z., Tomaro, R.F., and Witzeman, F.C., "*$Cobalt_{60}$*: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Solver," *Advances in Engineering Software*, Vol. 29, No. 3

7. Karypis, G., Schloegel, K. and Kumar, V., *ParMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering Library Version 1.0*. University of Minnesota, Department of Computer Science, Minneapolis, MN 55455, July 1997.

Figures

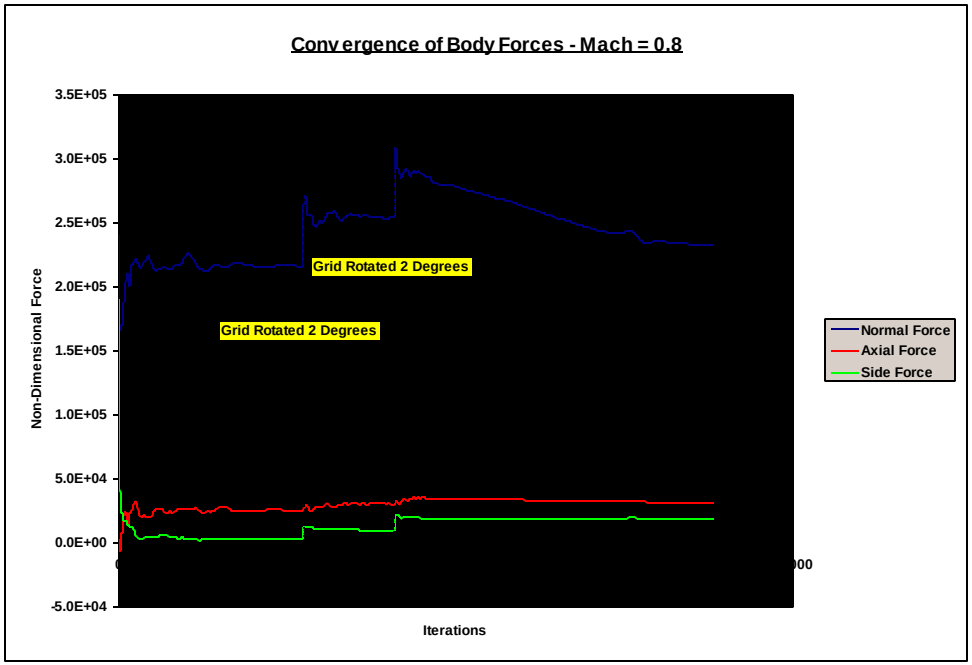


Figure 1. Body force convergence history. Mach = 0.8 case, Spalart-Allmaras turbulence model.

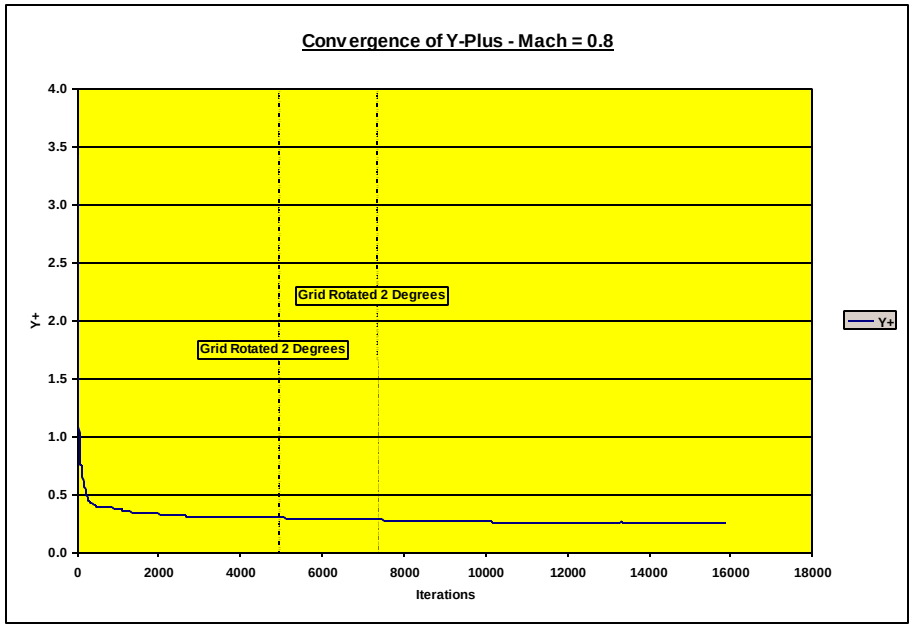


Figure 2. Y+ convergence history. Mach = 0.8 case, Spalart-Allmaras turbulence model.

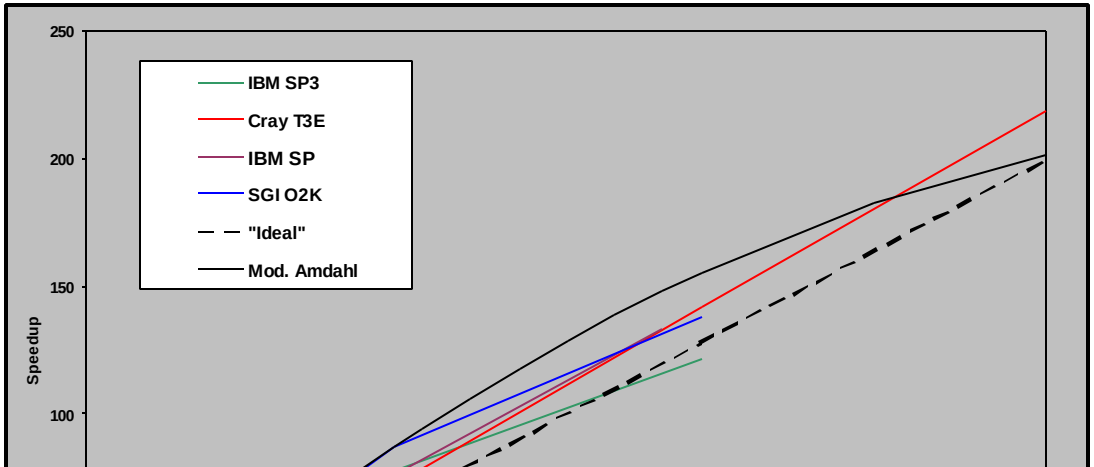


Figure 3. Parallel speed-up for 435,000 cell tetrahedral grid achieved on 3 different architectures.

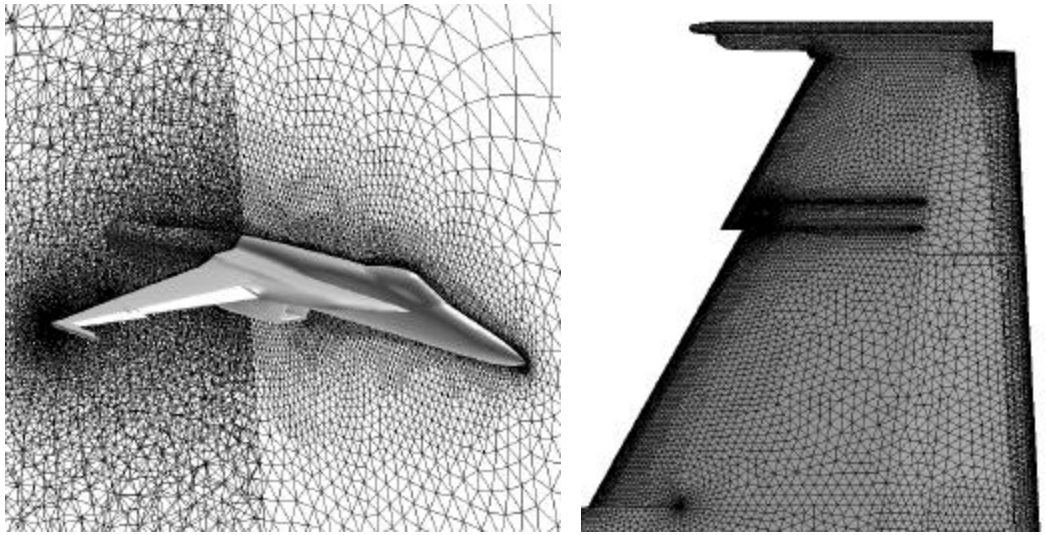


Figure 4. Volume grid and grid on wing surface.

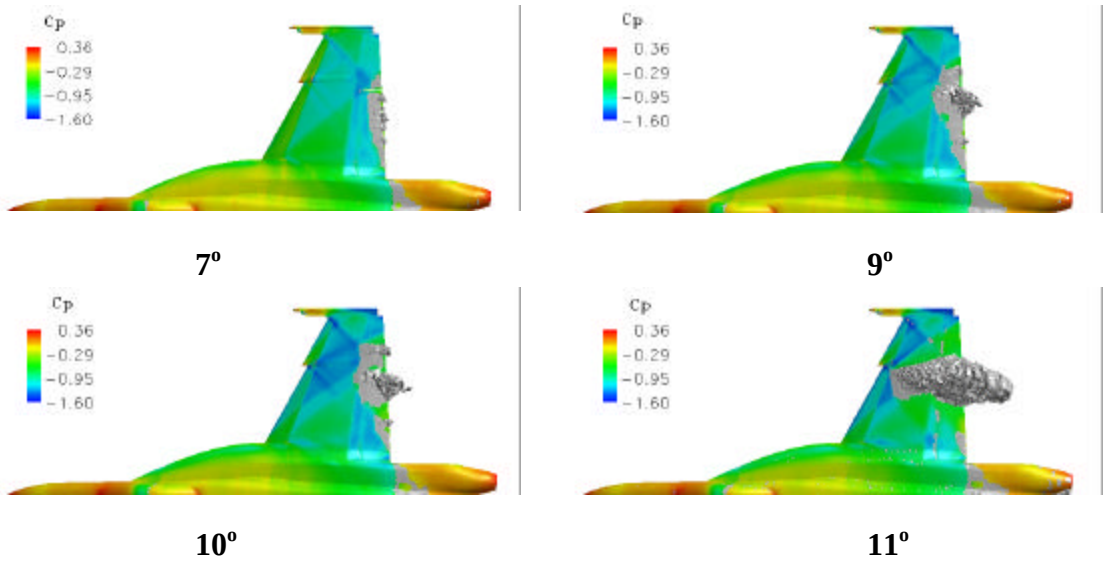
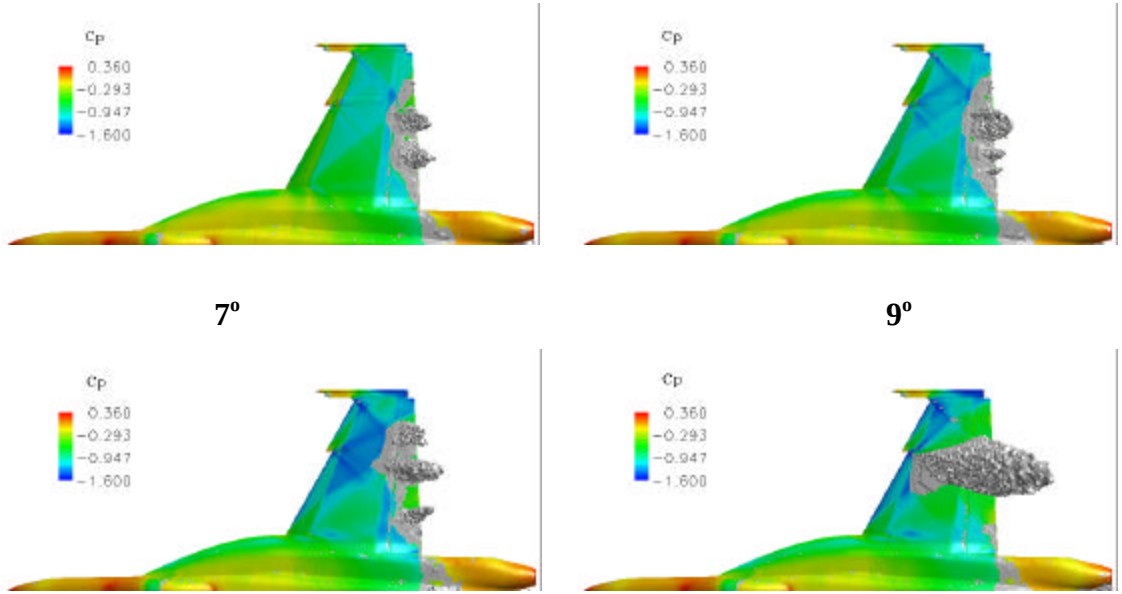


Figure 5. Top view of wing with surface contours of Cp. Spalart-Allmaras turbulence model. Thick trailing edge on wing. Mach = 0.9. Gray region is isosurface of zero-velocity flow (boundary of separated region).



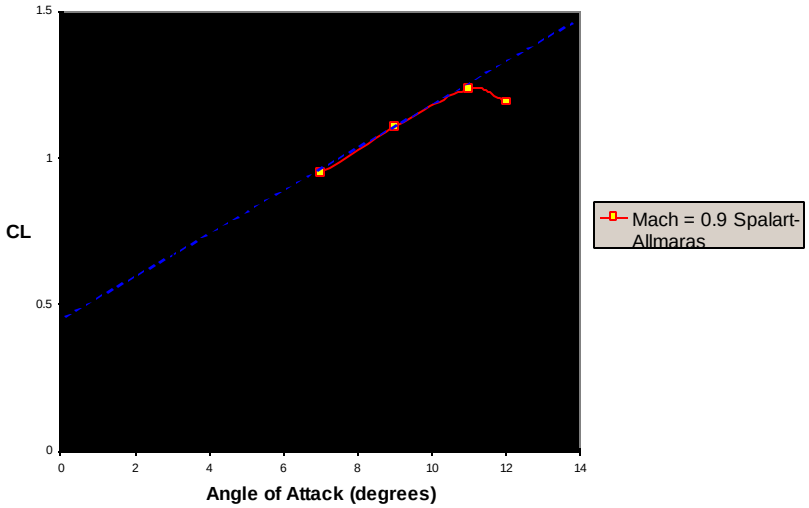
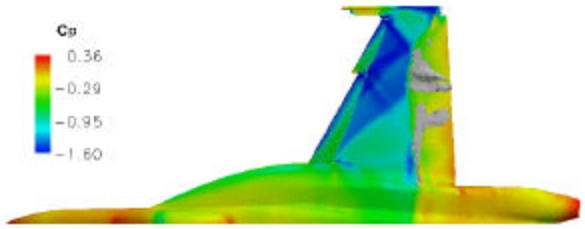
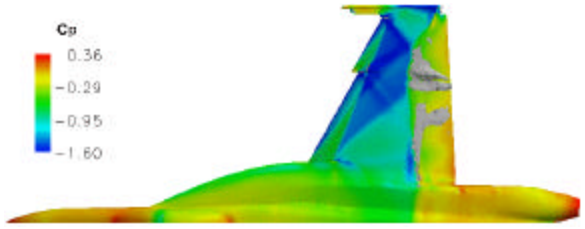


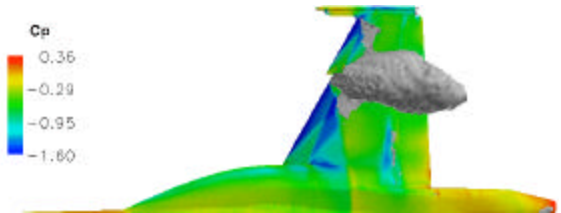
Figure 7. C_L vs. Angle of Attack curve for wing with infinitely thin trailing edge. Mach = 0.9.



6°



8°



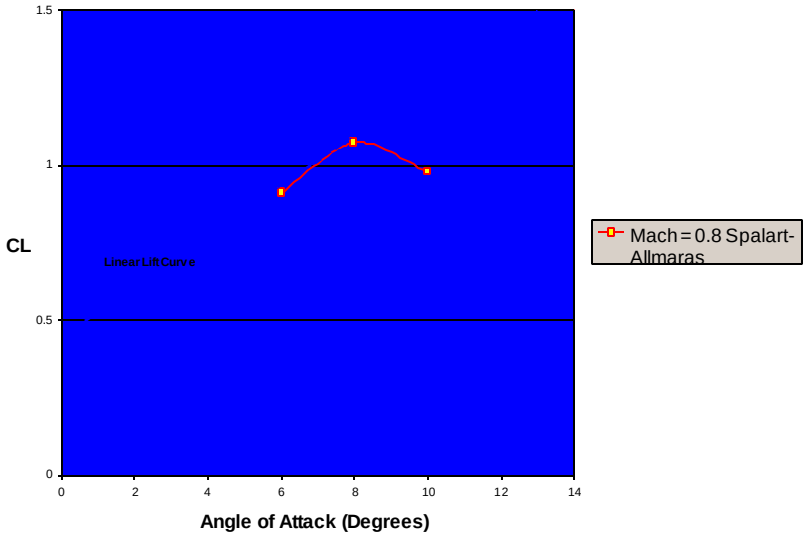
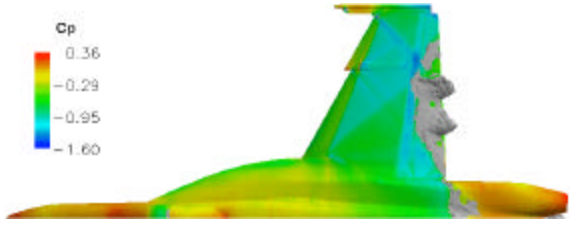
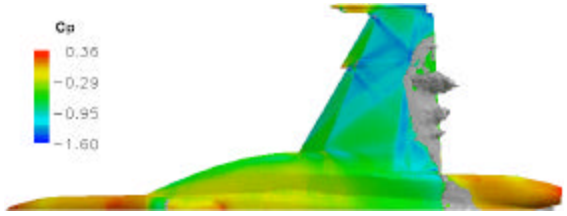


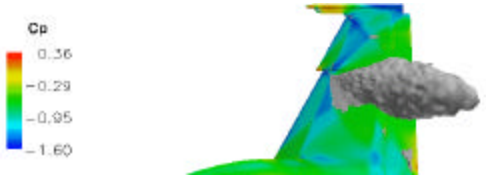
Figure 9. C_L vs. Angle of Attack curve for wing with infinitely thin trailing edge. Mach = 0.8.



7°



9°



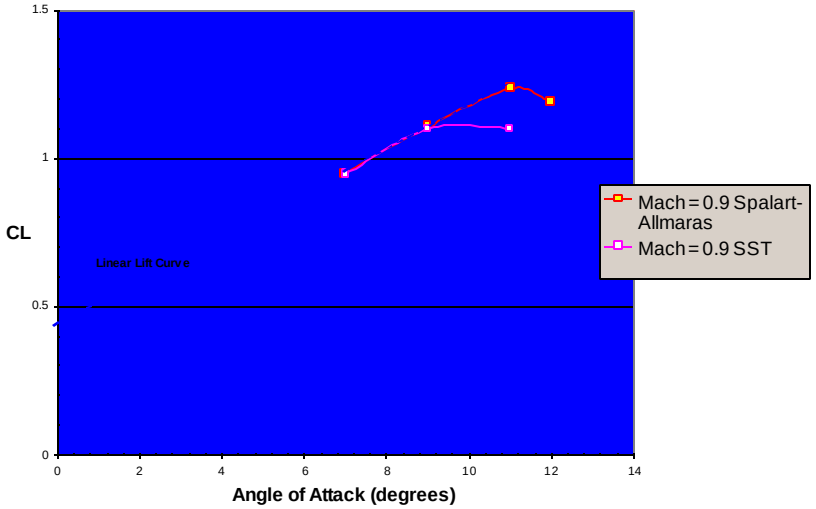


Figure 11. C_L vs. Angle of Attack curve for wing with infinitely thin trailing edge. Comparison of turbulence models.